Profiling Go Applications

Introduction to Profiling in Go Language

Pinal Dave
Developer

@pinaldave | blog.sqlauthority.com

Profiling Go Applications

Version Check

Version Check

This version was created by using:

- Go Language 1.20.1

Version Check

This course is 100% applicable to:

- Go Language 1.x

Relevant Notes

A note on GUI

- This course is built with GUI – Visual Studio Code; however, any other GUI or a standard notepad works perfectly fine

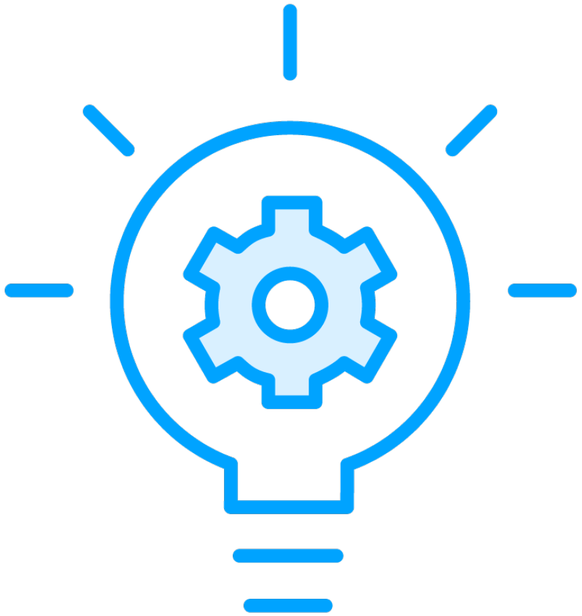
Intended Outcome

Learn to efficiently profile Go applications and web services to enhance their performance and user experience.

Commands:

- runtime/pprof
- net/http/pprof

Prerequisites



Skill

- Fundamentals of Go Language

Software

- Installation of Go Language
- Tool to edit code
- Command terminal

Overview

Introduction

Methods for profiling

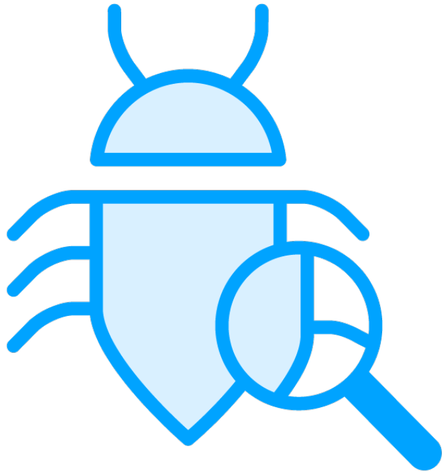
Predefined profiles

Profiling with benchmark

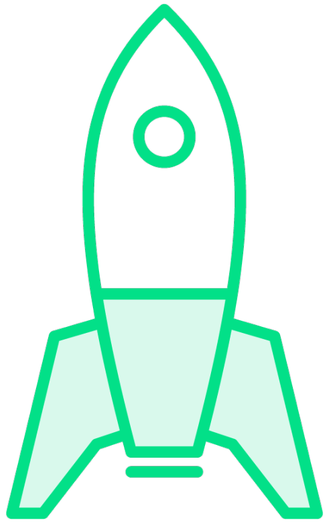
Profiling in Go Language

Analyze the complexity and costs to identify the expensive sections of Go programs

Why Profiling in Go Language?



Identification of bottlenecks

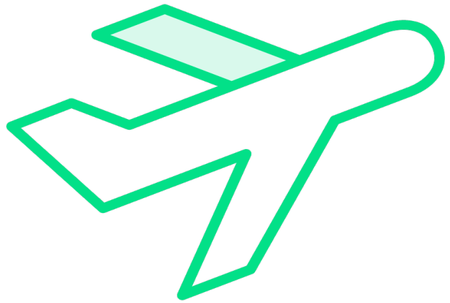


Performance optimization

Methods for Profiling

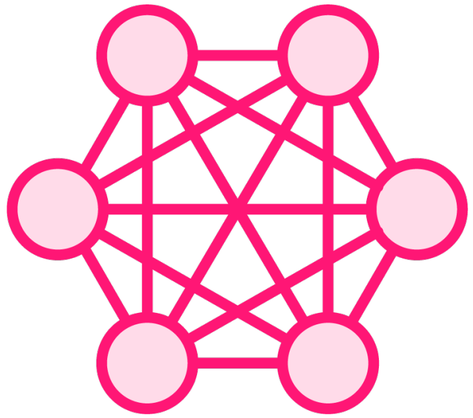
Built-in Benchmarking

go test



Runtime Profiling

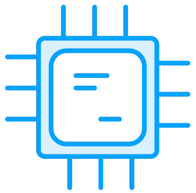
runtime/pprof



Active Web Profiling

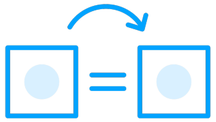
net/http/pprof

Predefined Profiles

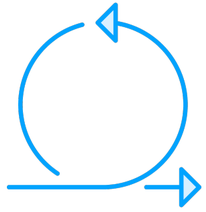


CPU

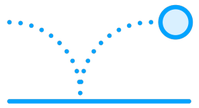
Heap/Memory



Block



Thread



Goroutine

Mutex

Demo

Profiling with benchmark

- go test

Demo

Profiling with benchmark

- pprof

Summary

Methods for profiling

- go test
- runtime/pprof
- net/http/pprof

Profiling with benchmark

Up Next:

Profiling with pprof